

**Cascading Style Sheets**

[CSS 2](#_Toc534976148)

[1 Зачем нужен CSS? 2](#_Toc534976149)

[2 Что значит CSS? 2](#_Toc534976150)

[3 В чем писать CSS? 2](#_Toc534976151)

[4 Просмотр CSS в браузере 2](#_Toc534976152)

[5 HTML-страница для примера 4](#_Toc534976153)

[6 Как вставить CSS в HTML? 6](#_Toc534976154)

[6.1 Встроенные стили 6](#_Toc534976155)

[6.2 Внутренние стили 7](#_Toc534976156)

[6.3 Внешние стили 9](#_Toc534976157)

[6.4 Задание 10](#_Toc534976158)

[7 Еще немного HTML 10](#_Toc534976159)

[8 Синтаксис CSS 15](#_Toc534976160)

[8.1 Псевдоклассы 16](#_Toc534976161)

[8.2 Единицы измерения 17](#_Toc534976162)

[8.3 Задание цвета 18](#_Toc534976163)

[8.4 Комментарии 18](#_Toc534976164)

[8.5 Каскадность стилей 18](#_Toc534976165)

[9 Основные свойства CSS 18](#_Toc534976166)

[9.1 Шрифт 19](#_Toc534976167)

[9.2 Абзац (paragraph) 21](#_Toc534976168)

[9.3 Цвет и фон 23](#_Toc534976169)

[9.4 Блочная структура 27](#_Toc534976170)

[9.5 Рамки 29](#_Toc534976171)

[Ресурсы 32](#_Toc534976172)

# CSS

## Зачем нужен CSS?

Сегодня CSS используется в неразрывной связке с HTML. Когда-то в HTML описывали и содержание, и оформление web-страницы. Теперь

* HTML отвечает за содержание (текст, рисунки) и разметку (абзацы, заголовки, ссылки, таблицы) страницы;
* CSS отвечает за оформление (шрифт, цвет, отступы, выравнивание).

В чем преимущества такого разделения?

* одинаковый стиль оформления для разных страниц (один CSS - много HTML страниц);
* экономия трафика - оформление загружается один раз, а не много раз для разных страниц;
* можно скачать готовые шаблоны оформления и подключить к своему сайту;
* разметку и дизайн могут делать разные люди;
* можно сделать несколько вариантов оформления и легко переключаться между ними.

## Что значит CSS?

**Cascading Style Sheets** (CSS читается «си-эс-эс» или «це-эс-эс») на русский принято переводить как **каскадные таблицы стилей**, хотя sheets точнее будет перевести как «листы» (текстовые файлы), а не «таблицы».

CSS можно встраивать прямо в HTML или (что лучше) сохранять в отдельный подключаемый файл.

## В чем писать CSS?

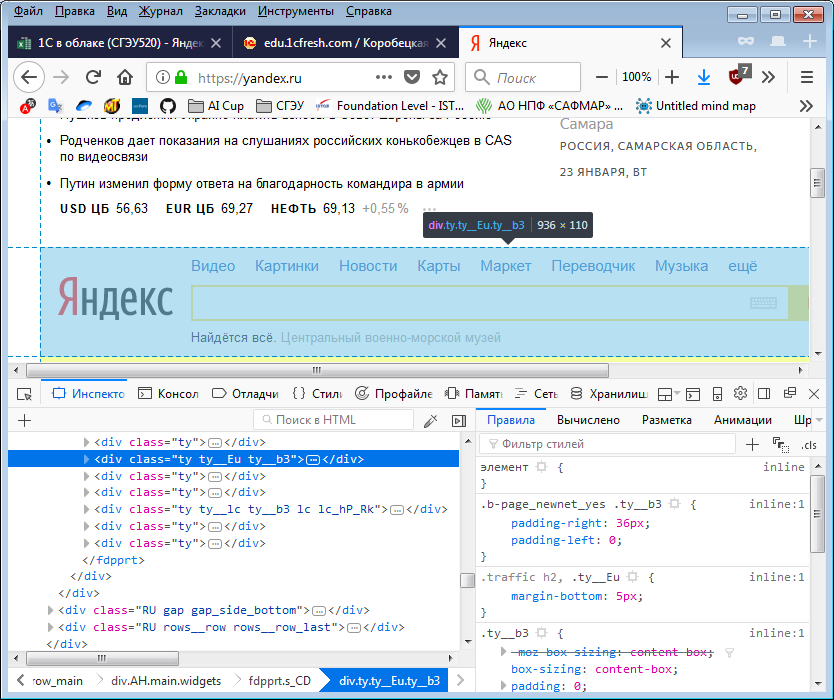
Как и HTML - в любом текстовом редакторе. Лучше всего с подсветкой кода, например, Notepad++.

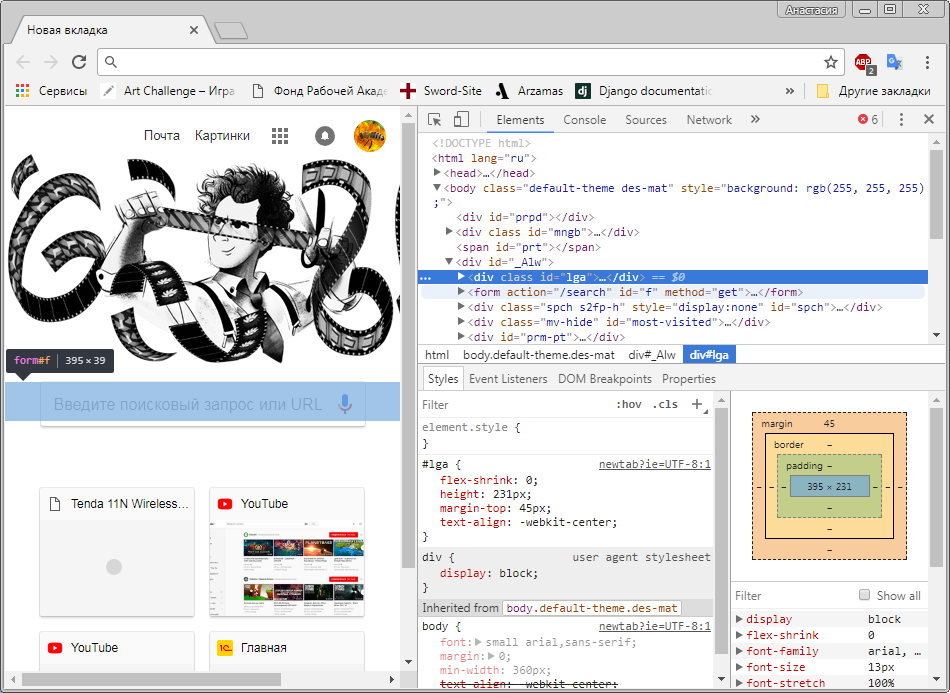
Для более профессиональной верстки подойдут пакеты типа WebStorm, Atom и др.

Наконец, есть онлайн-редакторы HTML и CSS, например, <https://jsfiddle.net/> Некоторые примеры ниже сделаны в <https://codepen.io/>

## Просмотр CSS в браузере

Все распространенные браузеры содержат встроенный просмотрщик и редактор CSS. Правый клик по странице – «Просмотр исходного кода элемента», стандартное сочетание клавиш Ctrl+Shift+I.





## HTML-страница для примера

Чтобы было к чему применять CSS, создадим простенькую страницу:

<html lang="ru">

<head >

<meta charset="utf-8">

<title>Моя первая страничка с CSS</title>

</head>

<body>

<!-- ШАПКА -->

<img src='logo.png'>

<h1> Название сайта </h1>

<!-- МЕНЮ -->

<nav>

<ul>

<li><a href="index.html">Главная</a>

<li><a href="about.html">О нас</a>

<li><a href="contacts.html">Контакты</a>

<li><a href="links.html">Ссылки</a>

</ul>

</nav>

<!-- ГЛАВНАЯ -->

<h2>Добро пожаловать!</h2>

<div>

Этот текст выделен в отдельный блок тегом div. Без CSS он выглядит как обычный абзац.

</div>

<p> Здесь размещается <span>основной текст</span> документа. Фраза «основной текст» выделена тегом span, но без CSS он ничем не отличается от обычного текста.</p>

<p> Еще один абзац.</p>

<p> Фраза «Съешь же еще этих мягких французских булок да выпей чаю» звучит немного странно, зато содержит почти все буквы русского алфавита, поэтому хорошо подходит для демонстрации шрифтов. Такие фразы называются панграммами, или разнобуквицей. В английском популярна панграмма ‘The quick brown fox jumps over the lazy dog’. Еще один английский вариант: ‘The five boxing wizards jump quickly’ (Пятеро боксирующих волшебников быстро подпрыгивают).</p>

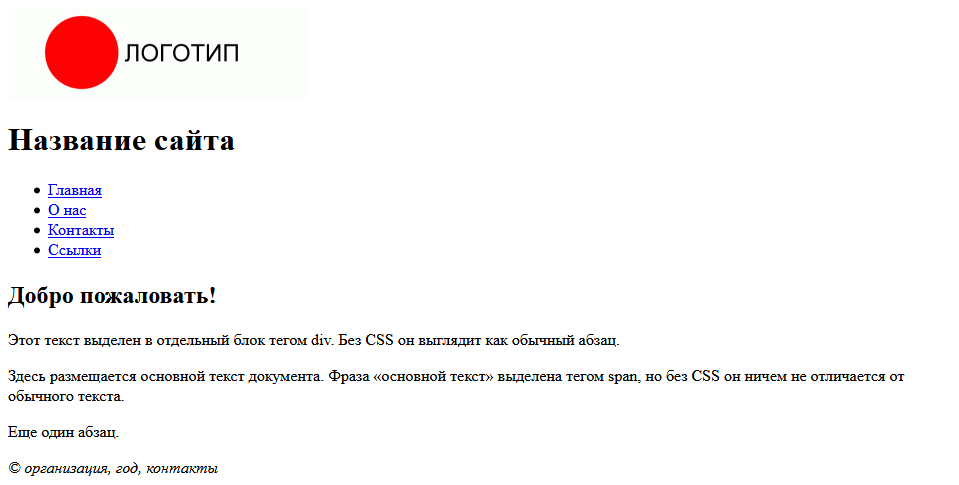
<!-- ПОДВАЛ -->

<address>&copy; организация, год, контакты </address>

</body>

</html>

В качестве логотипа используйте любую небольшую картинку.



## Как вставить CSS в HTML?

Есть три способа.

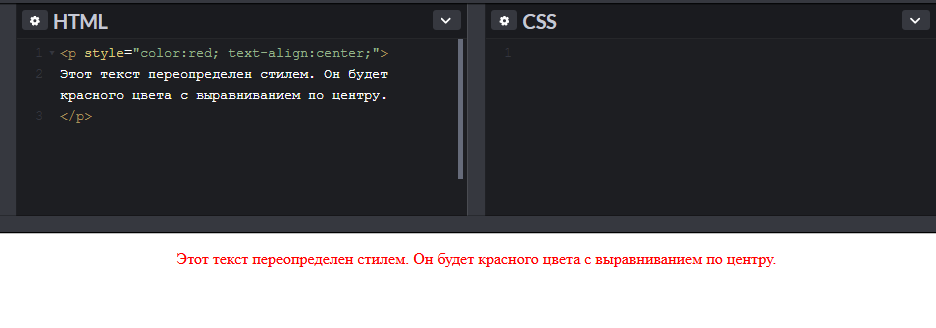
### Встроенные стили

При помощи дополнительного атрибута style мы можем определить нужные нам стилевые параметры в любом теге. Это самый легкий способ, и действует он в пределах лишь одного тега. Этот способ не слишком отличается, к примеру, от прямого описания внешнего вида при помощи тега <font>.

<p style="color:red; text-align:center;">

Этот текст переопределен стилем. Он будет красного цвета с выравниванием по центру.

</p>



### Внутренние стили

Действие таких стилей распространяется на всю страничку. При использовании этого метода описание стилей необходимо разместить в секции заголовка <head></head>.

Одинаковый стиль для разных тегов можно определить с помощью **класса**. В CSS класс начинается с точки, а дальше в фигурных скобках перечисляются его настройки:

<head> ....

<style type="text/css">

/\*стиль заголовка\*/

.header {

text-align :center;

font-size : 27pt;}

/\*стиль красного текста\*/

.red {

color : red;

}

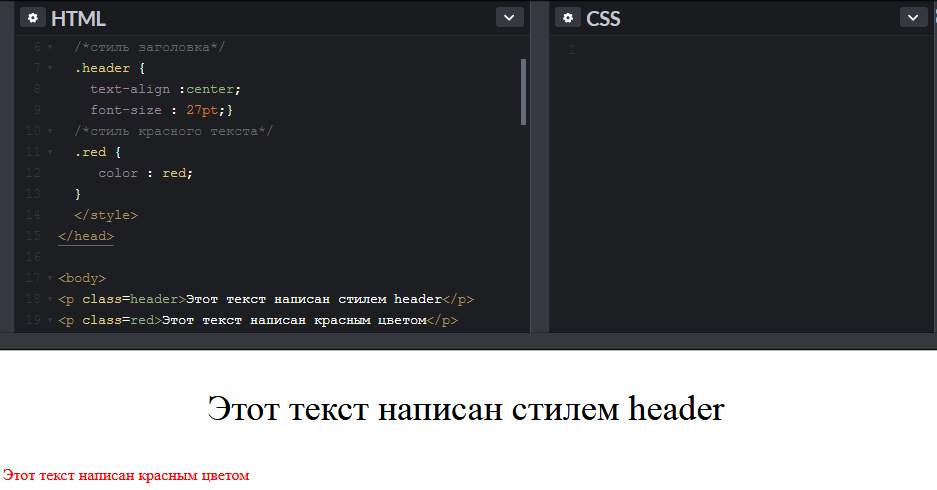
</style>

</head>

Теперь эти стили можно применять в любом месте html-кода. Для этого нужно указать созданный класс в атрибутах тега:

<p class=header>Этот текст написан стилем header</p>

<p class=red>Этот текст написан красным цветом</p>



Кроме определения новых классов можно **переопределять стандартные теги**: просто не нужно писать точку в начале. Например, тег <p>:

<style type="text/css">

p {

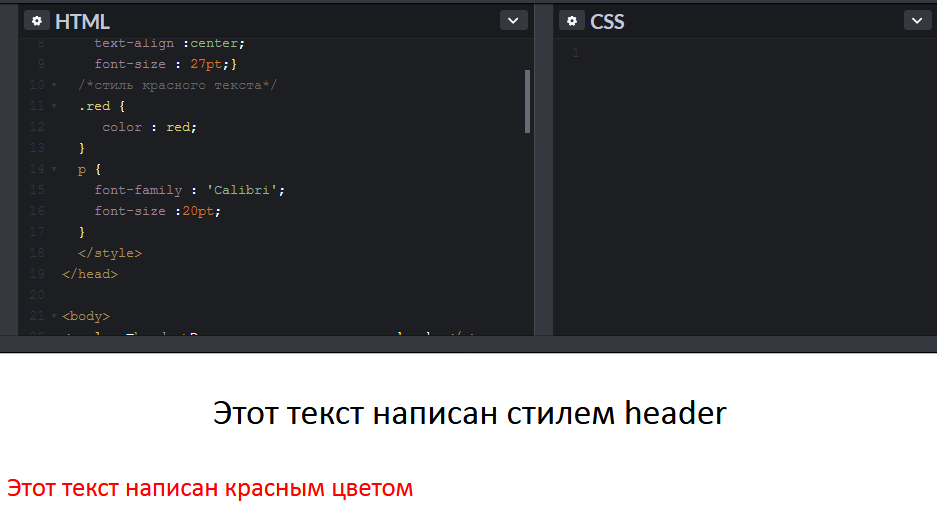
font-family : 'Calibri';

font-size :20pt;

}

</style>

Теперь весь текст, заключенный в теги <p></p>, будет выглядеть так, как определено данным стилем. Это очень удобно и позволяет легко адаптировать уже существующие странички к использованию стилей. Кроме того, не нужно писать лишних атрибутов class.



### Внешние стили

Находятся в отдельном файле.

Сначала создается CSS-файл с описанием всех нужных нам классов. Из предыдущего примера:

.header {

text-align : center;

font-size : 27pt;

}

.red {

color :red;

}

p {

font-family : 'Calibri';

font-size :20pt;

}

Этот файл нужно сохранить с расширением CSS, например, **mystyle.css**. Ссылка на него внедряется в документ при помощи тега <link>:

<head>

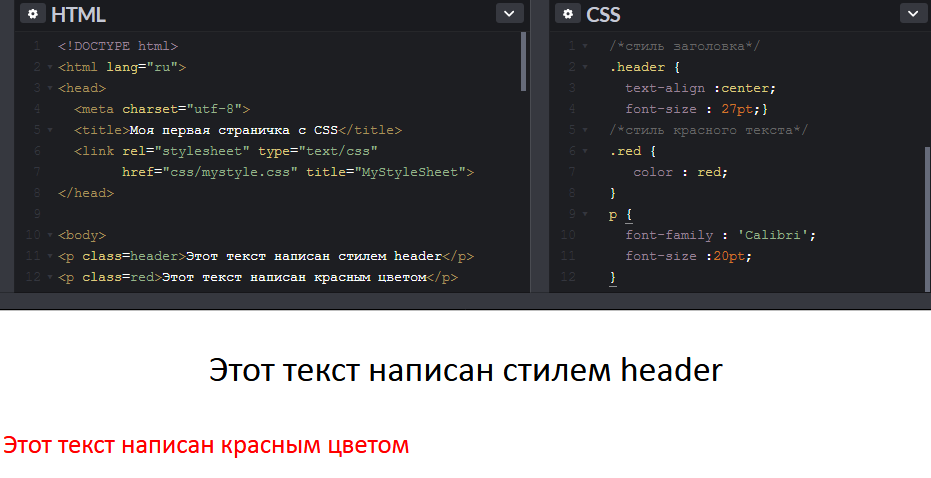
...

<link rel="stylesheet" type="text/css"

href="css/mystyle.css" title="MyStyleSheet">

</head>

Это самый удобный способ, и для основной таблицы стилей рекомендуется пользоваться именно им.



### Задание

Создайте **внешний CSS-файл** и подключите его к созданной ранее HTML-странице с примером.

Запишите в этом файле класс **.mystyle**. Задайте ему произвольный шрифт, размер и цвет. Подключите этот класс к тегу **div**.

Тегу **span** задайте через атрибут *style* синий цвет.

В заголовке страницы переопределите тег **img** так, чтобы он выравнивался по центру.

## Еще немного HTML

Для применения CSS нам потребуется еще два тега: <div> и <span>. Без CSS они не влияют на HTML, только позволяют выделить логические блоки на странице.

* **div**: служит для выделения блоков текста. Блоку можно настроить выравнивание, отступы, размещение на странице, заливку, шрифт и т.п. Блоки могут быть вложенными.
* **span**: служит для выделения текста внутри абзаца. Не может вкладываться один в другой. Для него можно настроить шрифт, цвет, заливку и т.п., но располагаться он всегда будет внутри текста.

Например, файл стилей:

/\* тело страницы\*/

body {

font-family: Arial, sans-serif;

}

/\*цитата\*/

.citation {

font-family: Times, serif;

font-style: italic;

margin: 10px 5px;

padding: 5px;

background-color: #FFFFF0;

}

/\*ключевые слова\*/

.keyword {

color: red;

font-size: 150%;

}

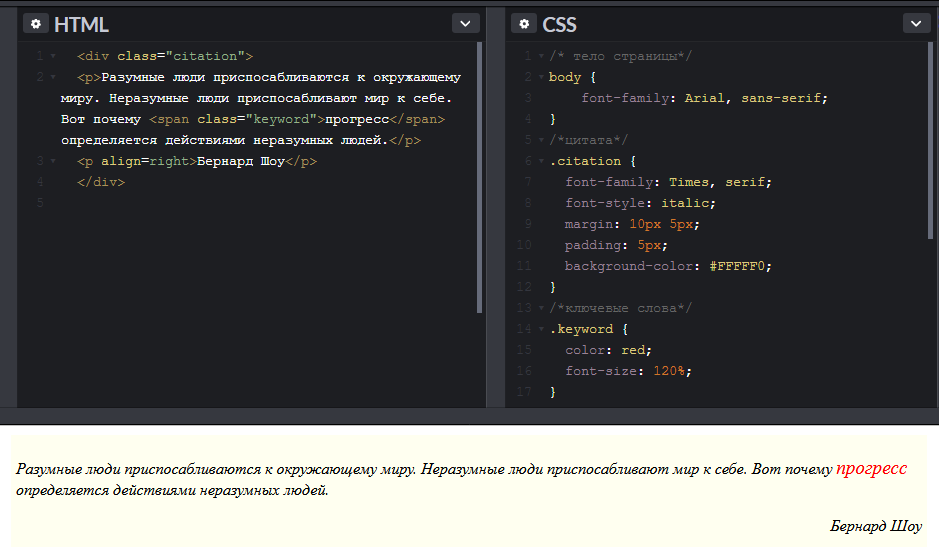
Текст на странице, оформленный с помощью стилей и тегов div и span:

<**div** class="citation">

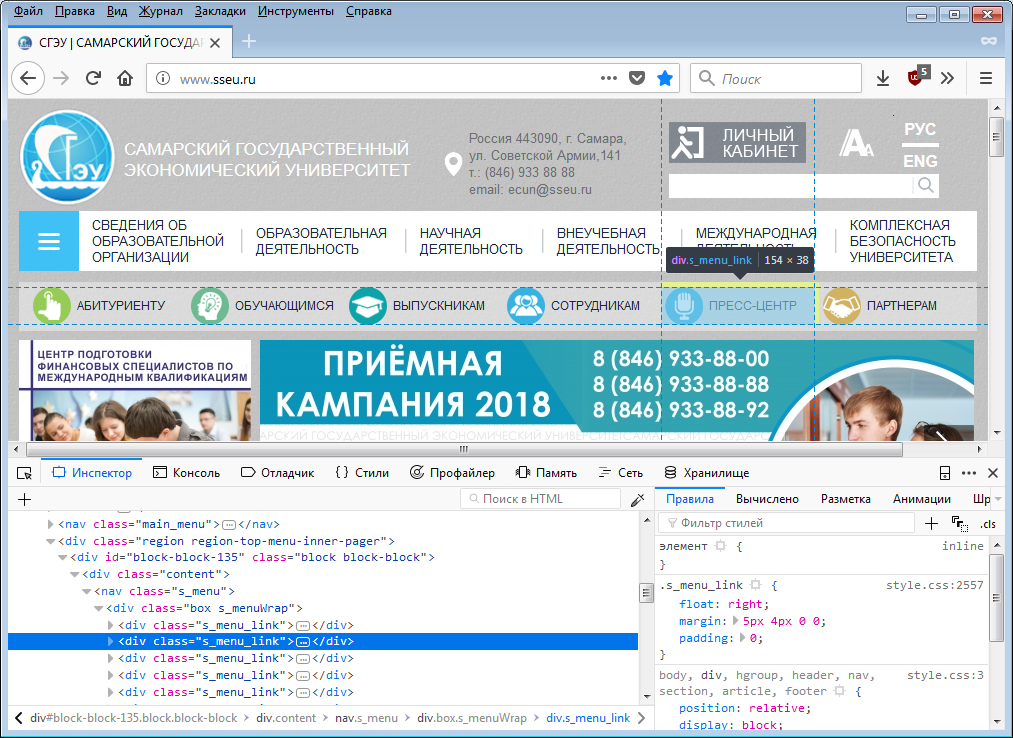
<p>Разумные люди приспосабливаются к окружающему миру. Неразумные люди приспосабливают мир к себе. Вот почему <**span** class="keyword">прогресс</**span**> определяется действиями неразумных людей.</p>

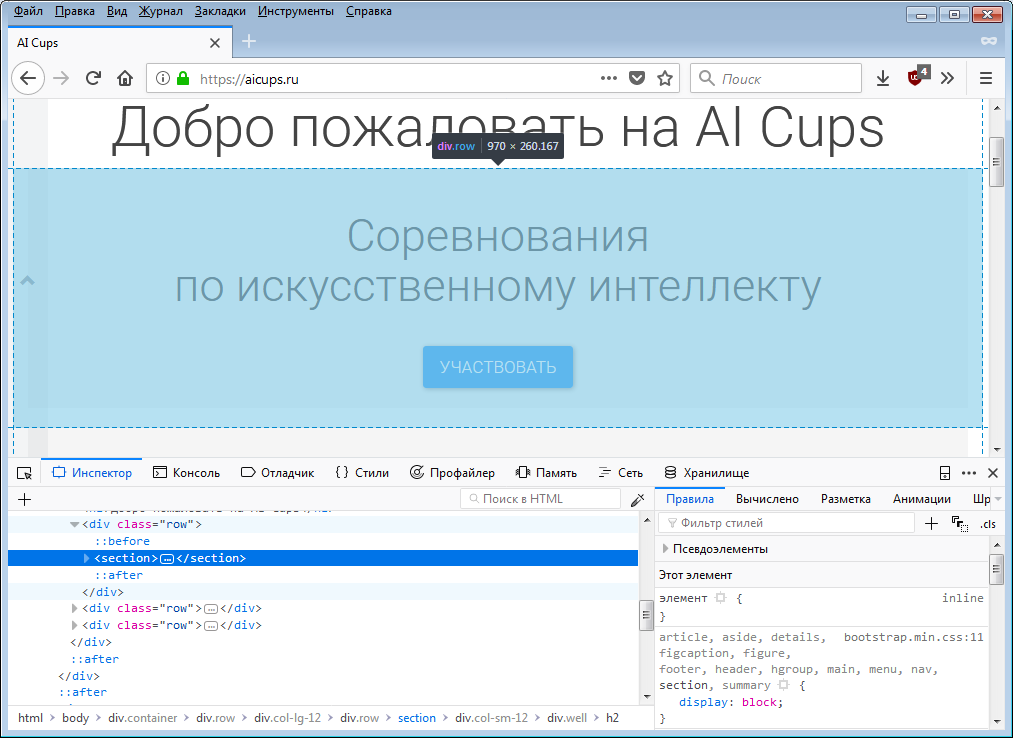
<p align=right>Бернард Шоу</p>

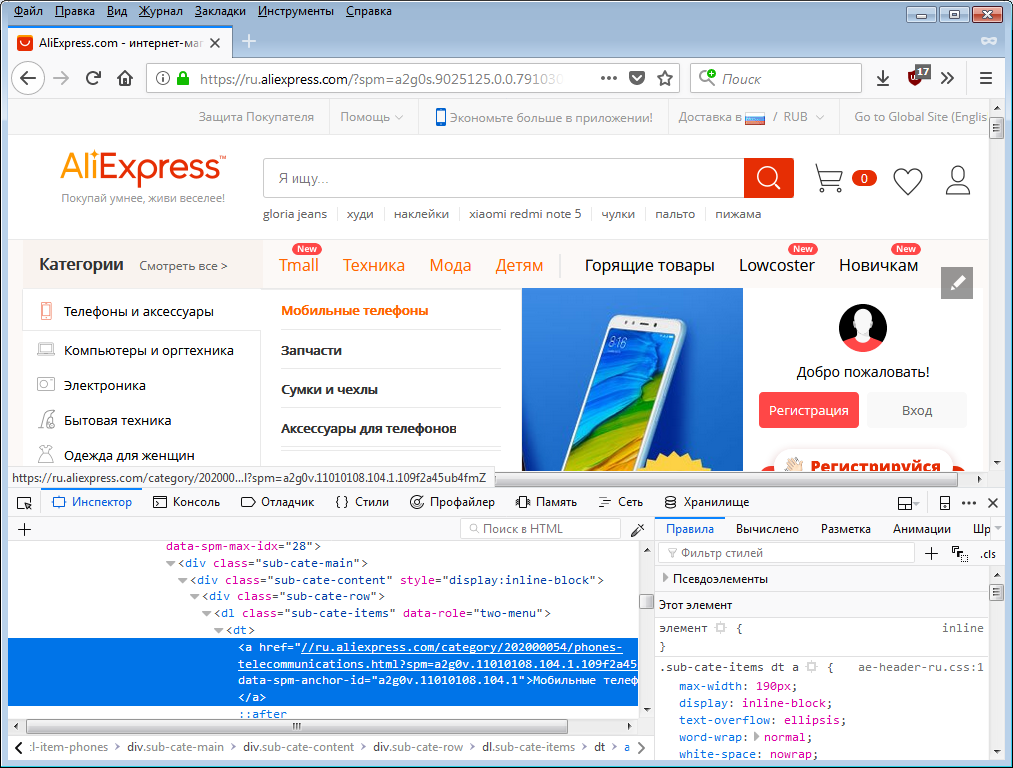
</**div**>



Сегодня любая web-страница состоит из множества вложенных тегов <div>.







### Задание

Добавьте внутрь блока **div** на странице еще один **div**, содержащий вашу фамилию и группу. Применится ли к нему стиль внешнего блока **div**?

Попробуйте задать тегу **span** стиль с выравниванием по центру. Повлияет ли это на абзац, в котором находится данный текст?

## Синтаксис CSS

Описание каждого класса делается при помощи конструкции, подобной этой:

.small {

font-size: 9pt;

}

Сначала указывается имя класса – оно может быть произвольным, но желательно все-таки давать осмысленное название. Далее, в фигурных скобках { } перечисляются параметры для данного класса. Параметры отделяются друг от друга точкой с запятой.

Имя класса начинается с точки и таким образом определяет ***универсальный класс***, т.е. такой, который может быть применен к любому тегу.

<p class=small>Накладываем стиль small на этот текст</p>

Существуют универсальные классы и так называемые ***теговые классы***:

p.small { font-size: 9pt; }

Класс, определенный таким образом, сработает только в том теге, для которого он предназначен, а для всех остальных будет проигнорирован.

Мы можем определять параметры не только для одного тега, но и сразу для нескольких. Для этого в определении стиля достаточно перечислить их через запятую:

p, td {

font-size: 9pt;

color:green;

}

Такой прием называется группировкой, и в данном случае мы определили и для <p>, и для <td> одинаковый размер и цвет текста.

В случае переопределения существующих тегов, в описании стиля можно указывать не все параметры, а лишь те из них, которые мы хотим изменить. Все остальные параметры примут значения по умолчанию, которые для разных тегов различны.

### Псевдоклассы

Действие ***псевдокласса*** распространяется не на весь текст, а лишь на его часть или в определенном состоянии. Например, эффект, при котором ссылки подчеркиваются лишь при наведении на них курсора:

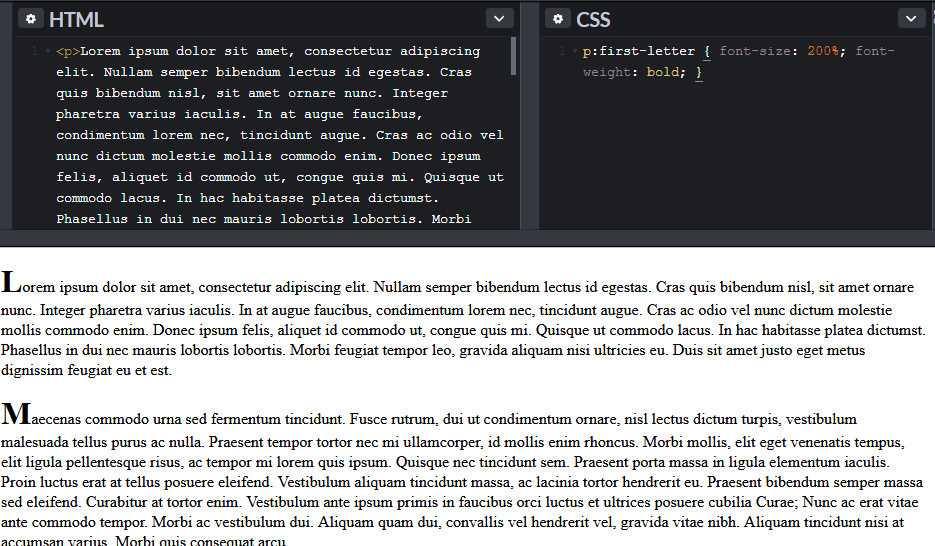
a { text-decoration: none; }

a:hover { text-decoration: underline;}

Первая строчка - это переопределение стандартного тега <a>, которое запрещает подчеркивать ссылки, а нижняя – это определение стиля для псевдокласса **hover** (наведение курсора мыши).

А вот и другой пример псевдокласса – определение буквицы (увеличенной первой буквы) в начале абзаца:

p:first-letter { font-size: 200%; font-weight: bold; }



Заметьте, что и в том, и в другом случае действие стиля распространяется либо на определенное состояние (пользователь собирается щелкнуть по ссылке), либо на фрагмент текста (изменяется только первая буква абзаца). В этом и заключается смысл псевдоклассов.

### Единицы измерения

В свойствах, которым требуется указание размеров, можно использовать несколько способов:

* относительный размер в процентах (%)
* относительный размер при помощи словесного описания (larger, smaller, xx-small, x-small, small, medium, large, x-large, xx-large)
* абсолютный размер в типографских единицах - в пунктах (pt), пиках (pc), пикселях (px), средней шириной буквы "m" (em), средней шириной буквы "x" (eх)
* абсолютный размер в стандартных единицах длины - размер может задаваться в сантиметрах (cm), миллиметрах (mm), дюймах (in), абсолютных пикселях (px)

### Задание цвета

Цвет для тех свойств, где это нужно, может быть определен одним из трех способов:

* при помощи названия цвета: yellow█, red█, magenta█, grey█;
* шестнадцатеричным заданием цвета в формате #RRGGBB: #00ff00█, #883490█, #ffffff (белый);
* десятичным заданием составляющих цвета в формате rgb: rgb(0,0,255)█, rgb(23,100,78)█.

Таблицы цветов с названиями и обозначениями в HEX-формате можно легко найти в интернете.

### Комментарии

Как и в любом достаточно сложном языке, при создании таблицы стилей можно пользоваться комментариями. Их формат аналогичен классическому C:

/\* Этот текст является комментарием. Поясните, что за стиль вы создали \*/

Золотое правило - чем понятнее названа переменная (в данном случае имя класса), тем меньше комментариев необходимо.

### Каскадность стилей

Каскадность заключается в том, что стили могут **переопределяться**, т.е. один и тот же класс можно определить несколько раз. Приведенный выше список способов внедрения стилей соответствует порядку переопределения: наибольший приоритет имеют встроенные в тег стили, наименьший – стили по умолчанию, назначаемые браузером.

Например, мы определили во внешнем стилевом файле, что текст в теге <p> должен быть написан при помощи шрифта высотой 10 пунктов.

/\* во внешнем файле \*/

p {

font-size: 10pt;

}

Но если в заголовке странички мы дополнительно укажем, что тот же тег <p> должен быть написан шрифтом в 12 пунктов, то текст будет выведен именно 12 кеглем – т.е. стиль в заголовке странички переопределил (перекрыл) стиль во внешнем файле.

<style>

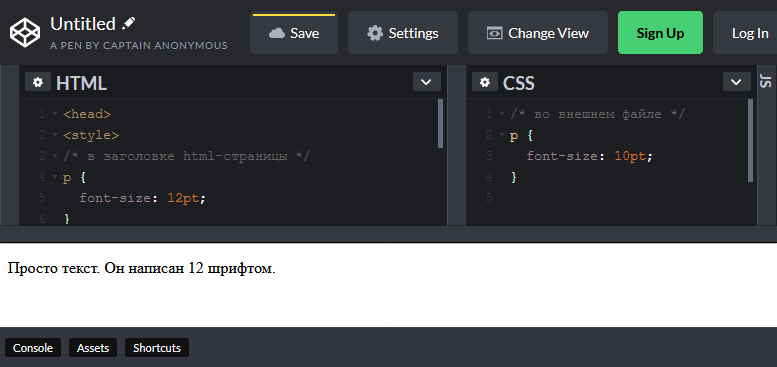
/\* в заголовке html-страницы \*/

p {

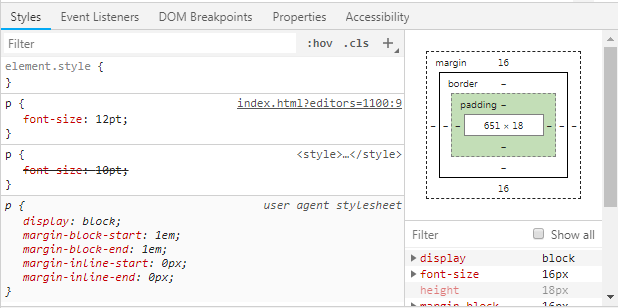
font-size: 12pt;

}

</style>



При этом браузер показывает в Инспекторе стилей переопределенные настройки как зачеркнутые.



### Задание

Используя справочную информацию из следующего раздела, настройте стили всем тегам на странице-примере.

Меню (nav) возьмите в рамку, сделайте однотонную заливку и разместите по правому краю страницы. Ссылки в меню не должны быть подчеркнутыми. Пункты меню должны подсвечиваться другим цветом при наведении мыши (псевдокласс hover).

Подвал (address) разместите по нижнему краю страницы, с выравниванием по центру, уменьшенным шрифтом, курсивом.

Выделите тегами с одинаковым стилем все панграммы в тексте. Само слово «панграмма» выделите жирным курсивом.

Остальные теги настройте по своему усмотрению.

Сгенерируйте с помощью <https://fish-text.ru/> еще четыре абзаца текста и добавьте их на страницу.

## Основные свойства CSS

Дальнейшая информация носит характер справочника. Вы можете прочитать все подряд, а можете обращаться к нужным разделам в любом порядке. Заданий там нет.

Рекомендуется внимательно ознакомиться с разделом 9.4 Блочная структура.

В конце приведены ссылки на дополнительные ресурсы.

### Шрифт

font-family

font-style

font-variant

font-weight

font-size

font

#### Семейство шрифта [font-family]

Свойство font-family указывает приоритетный список шрифтов. Если первый шрифт списка не установлен на компьютере, с которого выполняется доступ к сайту, ищется следующий шрифт списка, пока не будет найден подходящий.

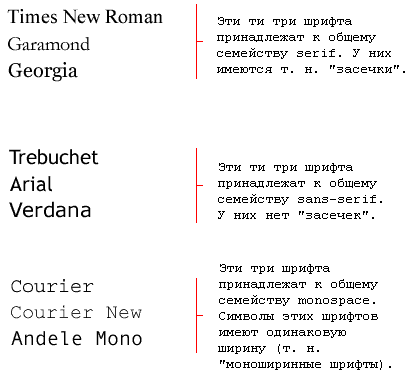
Используются два типа имён шрифтов:

**Family-name**

Примеры family-name (часто называемое просто "шрифт") это "Arial", "Times New Roman" или "Tahoma". Кавычки можно не ставить, если в названии шрифта нет пробелов

**Generic family**

Группа шрифтов, имеющих характерные общие черты. Пример – sans-serif, набор шрифтов без "засечек ".



При указании шрифтов для вашего web-сайта вы, естественно, начинаете с предпочтительного шрифта, а затем перечисляете альтернативные. Рекомендуем в конце списка указывать родовое имя. Тогда страница, как минимум, будет отображена шрифтом того же семейства, если отсутствуют все специфицированные конкретные шрифты.

h1 {font-family: arial, verdana, sans-serif;}

h2 {font-family: "Times New Roman", serif;}

Заголовки <h1> будут отображаться шрифтом "Arial". Если он не установлен на пользовательской машине, будет использоваться "Verdana". Если недоступны оба шрифта, для показа заголовков будет использован шрифт семейства **sans-serif**.

Обратите внимание, что имя шрифта "Times New Roman" содержит пробелы, поэтому указано в двойных кавычках.

#### Стиль шрифта [font-style]

Свойство font-style определяет **normal** (обычный), **italic** (*курсив*) или **oblique** (наклонный, принудительный наклон букв для шрифтов, не поддерживающих *italic*). В примере все заголовки <h2> будут показаны курсивом italic.

h1 {font-family: arial, verdana, sans-serif;}

h2 {font-family: "Times New Roman", serif; **font-style: italic;**}

#### Вес шрифта [font-weight]

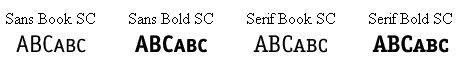
Свойство font-weight описывает, насколько толстым, или "тяжёлым", должен отображаться шрифт. Шрифт может быть **normal** или **bold**. Некоторые браузеры поддерживают и числовые значения 100-900 (в сотнях) для описания веса шрифта.

p {font-family: arial, verdana, sans-serif;}

td {font-family: arial, verdana, sans-serif; **font-weight: bold;**}

#### Вариант шрифта [font-variant]

Свойство font-variant используется для выбора между вариантами **normal** и **small-caps** (малые прописные). Шрифт **small-caps** использует малые заглавные буквы (upper case) вместо букв нижнего регистра. Примеры:



Если font-variant имеет значение **small-caps**, а шрифт small-caps недоступен, браузер, скорее всего, отобразит текст буквами верхнего регистра.

h1 {font-variant: small-caps;}

h2 {font-variant: normal;}

#### Размер шрифта [font-size]

Размер шрифта в любых единицах измерения:

h1 {font-size: **30px**;}

h2 {font-size: **12pt**;}

h3 {font-size: **120%**;}

p { font-size: **1em;**}

#### Сокращённая запись [font]

Используя сокращенную запись font, можно указывать все свойства шрифта в одну строку. Например, полная запись для тега <p>:

p {

font-style: italic;

font-weight: bold;

font-size: 30px;

font-family: arial, sans-serif;

}

Используя сокращённую запись, код можно упростить:

p {

font: italic bold 30px arial, sans-serif;

}

Порядок свойств font:

font-style | font-variant | font-weight | font-size | font-family

### Абзац (paragraph)

text-indent

text-align

text-decoration

letter-spacing

text-transform

#### Отступы [text-indent]

Свойство text-indent позволяет выделить абзац с помощью установки **отступа первой строки**. В примере отступ первой строки 30px применяется ко всем <p>:

p {

text-indent: 30px;

}

#### Выравнивание текста [text-align]

CSS-свойство text-align соответствует атрибуту тега <p>, используемому в старых версиях HTML. Текст может быть выровнен **left**, **right**, **center** или **justify**.

В примере текст заголовочных ячеек таблицы <th> выравнивается вправо, а в ячейках данных <td> - по центру, а обычные абзацы - justify:

th {

text-align: **right;**

}

td {

text-align: **center;**

}

p {

text-align: **justify;**

}

#### Обрамление текста [text-decoration]

Свойство text-decoration позволяет добавлять различные эффекты с линиями. Например, можно подчеркнуть текст, провести линию ~~по тексту~~ или даже  и т.д. В примере <h1> подчёркнуты, <h2> - имеют черту над текстом, а <h3> - перечёркнуты.

h1 {

text-decoration: **underline**; /\*нижняя линия\*/

}

h2 {

text-decoration: **overline**; /\*верхняя линия\*/

}

h3 {

text-decoration: **line-through**; /\*линия насквозь\*/

}

#### Интервал между буквами [letter-spacing]

Пустое место между всеми буквами (разреженный шрифт). Например:

h1 {

letter-spacing: **6px;**

}

p {

letter-spacing: **3px;**

}

#### Регистр символов [text-transform]

Например, слово "headline" можно показать как "HEADLINE" или "Headline". Имеются четыре возможных значения text-transform:

**capitalize**

Делает заглавной первую букву каждого слова. Например: "john doe" станет "John Doe".

**uppercase**

Конвертирует все символы в верхний регистр. Например: "john doe" станет "JOHN DOE".

**lowercase**

Конвертирует все символы в нижний регистр. Например: "JOHN DOE" станет "john doe".

**none**

Трансформации нет - текст отображается так же, как в исходном HTML-коде.

Для примера мы используем список имён. Все имена выделены с помощью <li> (list-item). Давайте капитализируем все имена и отобразим все заголовки верхним регистром.

h1 {

text-transform: **uppercase;**

}

li {

text-transform: **capitalize;**

}

### Цвет и фон

color

background-color

background-image

background-repeat

background-attachment

background-position

background

#### Цвет переднего плана [color]

Свойство color описывает основной цвет переднего плана элемента (обычно цвет шрифта, или цвет линии).

Например, мы хотим сделать все заголовки документа тёмно-красными.

h1 {

**color: #990000;**

}

Цвета можно указывать как шестнадцатеричные значения, как в примере (#ff0000), либо вы можете использовать названия цветов ("red") или rgb-значения (rgb(255,0,0)).

#### Цвет фона [background-color]

В элементе <body> размещается всё содержимое HTML-документа. Таким образом, для изменения цвета фона всей страницы свойство background-color нужно применить к элементу <body>.

Вы можете также применять это свойство к другим элементам, в том числе - к заголовкам и тексту. В следующем примере различные цвета фона применяются к элементам <body> и <h1>.

body {

**background-color: #FFCC66;**

}

h1 {

color: #990000;

**background-color: #FC9804;**

}

#### Фоновые изображения [background-image]

Для вставки рисунка в качестве фонового изображения web-страницы просто примените свойство background-image в тэге <body> и укажите местоположение рисунка.

body {

background-color: #FFCC66;

**background-image: url("butterfly.gif");**

}

Обратите внимание, что мы специфицируем место, где находится файл как **url("butterfly.gif")**. Это означает, что он находится в той же папке, что и таблица стилей. Разумеется, можно ссылаться и на файлы изображений в других папках, используя,например, **url("/images/butterfly.gif"),** или на файлы в сети, указывая полный адрес: **url("http://www.html.net/butterfly.gif")**.

#### Повторение фонового изображения [background-repeat]

Управляет заполнением фонового рисунка на пространстве страницы. Возможные значения:

|  |  |
| --- | --- |
| Значение | Описание |
| background-repeat: repeat-x | Рисунок повторяется по горизонтали |
| background-repeat: repeat-y | Рисунок повторяется по вертикали |
| background-repeat: repeat | Рисунок повторяется по горизонтали и вертикали |
| background-repeat: no-repeat | Рисунок не повторяется |

Например, для отмены повторения/мультипликации фонового рисунка мы должны записать такой код:

body {

background-color: #FFCC66;

background-image: url("butterfly.gif");

**background-repeat: no-repeat;**

}

#### Привязка фонового изображения при прокрутке [background-attachment]

Свойство background-attachment определяет, фиксируется ли фоновый рисунок, или прокручивается вместе с содержимым страницы.

|  |  |
| --- | --- |
| Значение | Описание |
| Background-attachment: scroll | Изображение прокручивается вместе со страницей – разблокировано. |
| Background-attachment: fixed | Изображение привязано. При прокручивании страницы изображение не будет двигаться. |

Например, следующий код фиксирует изображение и оно не будет повторяться (на фоне всегда будет одна картинка, а текст будет пролистываться поверх нее).

body {

background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

**background-attachment: fixed;**

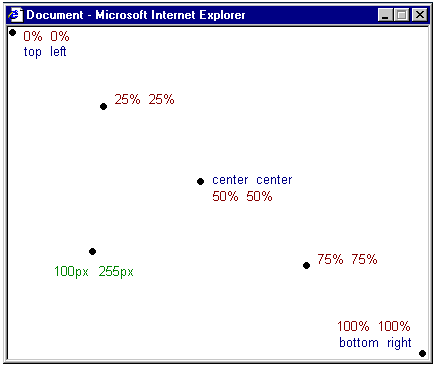
}

#### Расположение фонового рисунка [background-position]

По умолчанию фоновый рисунок позиционируется в левом верхнем углу экрана. Свойство background-position позволяет изменять это значение по умолчанию, и фоновый рисунок может располагаться в любом месте экрана.

Есть много способов установить значение background-position. Тем не менее, все они представляют собой набор координат. Например, значение '100px 200px' располагает фоновый рисунок на 100px слева и на 200px сверху в окне браузера.

Координаты можно указывать в процентах ширины экрана, в фиксированных единицах (пикселы, сантиметры, и т. п.), либо вы можете использовать слова top, bottom, center, left и right.



В таблице дано несколько примеров.

|  |  |
| --- | --- |
| Значение | Описание |
| background-position: 2cm 2cm | Рисунок расположен на 2 cm слева и на 2 cm сверху |
| background-position: 50% 25% | Рисунок расположен по центру и на четверть экрана сверху |
| background-position: top right | Рисунок расположен в правом верхнем углу страницы |

В примере кода фоновое изображение располагается в правом нижнем углу экрана:

body {

background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

background-attachment: fixed;

**background-position: right bottom;**

}

#### Сокращённая запись [background]

С помощью background вы можете сжимать несколько свойств и записывать стили в сокращённом виде, что облегчает чтение таблиц.

Например:

background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

background-attachment: fixed;

background-position: right bottom;

одной строкой кода:

background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;

Порядок свойств (можно пропускать, поэтому в [ ]):

[background-color] [background-image] [background-repeat] [background-attachment] [background-position]

Если свойство отсутствует, оно автоматически получает значение по умолчанию. Например, если background-attachment и background-position нет в данном примере:

background: #FFCC66 url("butterfly.gif") no-repeat;

Этим двум неспецифицированным свойствам будут присвоены значения по умолчанию - scroll и top left.

### Блочная структура

В концепции CSS любую часть документа можно представить в виде **блока** (box):

**Содержимое блока (box content)**

Поля (margin)

Отступы (padding)

Рамка (border)

top

bottom

right

left

текст снаружи блока или края документа

Блоки могут вкладываться один в другой. Например, блок в тегах <body> </body> содержит внутри себя все остальные блоки. Картинка <img> и тег <div></div> - тоже блоки. А вот теги <a></a>, <font></font> и <span></span> блоками не являются (не начинаются с нового абзаца).

#### Поля [margin]

У элемента есть четыре стороны: right, left, top и bottom. Поля margin это расстояние от каждой стороны с до соседних элементов (или краёв документа).

В качестве первого примера мы разберёмся, как определить поля самого документа, т. е. элемента <body>. На иллюстрации показано, какие поля нам нужны.

|  |  |
| --- | --- |
| Examples of margins | body {  margin-top: 100px;  margin-right: 40px;  margin-bottom: 10px;  margin-left: 70px;  }  Или:  body {  margin: 100px 40px 10px 70px;  } |

#### Отступы (внутренние поля) [padding]

Определяет внутреннее расстояние между рамкой и содержимым элемента. Использование padding можно показать на простом примере, где все заголовки имеют цветной фон. Определяя заполнение для заголовков, вы устанавливаете величину поля вокруг текста каждого заголовка:

h1 {

background: yellow;

**padding: 20px 20px 20px 80px;**

}

h2 {

background: orange;

**padding-left:120px;**

}

#### Ширина [width]

div.box {

width: 200px;

border: 1px solid black;

background: orange;

}

#### Высота [height]

Обратите внимание, в предыдущем примере высота блока устанавливается его содержимым. Высоту элемента можно определить свойством height. В качестве примера попытаемся создать блок с высотой 500px:

div.box {

height: 500px;

width: 200px;

border: 1px solid black;

background: orange;

}

#### Поплавок [float]

Определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с других сторон (не путайте с выравниванием текста text-align, которое действует внутри блока). Когда значение свойства float равно none, элемент выводится на странице как обычно, при этом допускается, что одна строка обтекающего текста может быть на той же линии, что и сам элемент.

Возможные значения:

**left**

Выравнивает элемент по левому краю, а все остальные элементы, вроде текста, обтекают его по правой стороне.

**right**

Выравнивает элемент по правому краю, а все остальные элементы обтекают его по левой стороне.

**none**

Обтекание элемента не задается.

**inherit**

Наследует значение вышестоящего класса в каскаде.

Например, картинка по левому краю:

img.leftside {

margin: 10px 10px 10px 10px;

border: 1px solid red;

**float: left**;

}

### Рамки

border-width

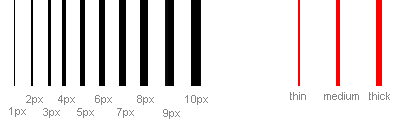
border-color

border-style

border

#### Толщина рамки [border-width]

Толщина рамки определяется свойством border-width, которое может иметь значения thin, medium и thick, или числовое значение в пикселах.



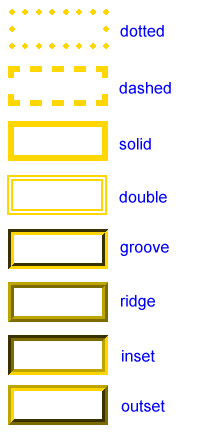
#### Цвет рамки [border-color]

Свойство border-color определяет цвет рамки. Цвет задается как и везде.

#### Стиль рамки [border-style]

Существуют различные типы рамок. Ниже показаны восемь типов рамки. Все примеры показаны цветом gold и толщиной thick.

Значения none или hidden могут использоваться, если вы не хотите отображать рамку.



Три рассмотренных выше свойства можно объединить в каждом элементе и, соответственно, устанавливать разные рамки. Для иллюстрации взглянем на документ, где определены разные рамки для <h1>, <h2>, <ul> и <p>:

/\* толстая штриховая рамка золотистого цвета\*/

h1 {

border-width: thick;

border-style: dotted;

border-color: gold;

}

/\*красная объемная рамка толщиной 20px\*/

h2 {

border-width: 20px;

border-style: outset;

border-color: red;

}

/\*пунктирная рамка синего цвета толщиной 1px\*/

p {

border-width: 1px;

border-style: dashed;

border-color: blue;

}

/\*тонкая сплошная оранжевая\*/

ul {

border-width: thin;

border-style: solid;

border-color: orange;

}

Можно также установить специальные свойства для верхнего, нижнего, правого и левого края рамки:

h1 {

/\*сверху красная\*/

border-top-width: thick;

border-top-style: solid;

border-top-color: red;

/\*снизу синяя\*/

border-bottom-width: thick;

border-bottom-style: solid;

border-bottom-color: blue;

/\*справа зеленая\*/

border-right-width: thick;

border-right-style: solid;

border-right-color: green;

/\*слева оранжевая\*/

border-left-width: thick;

border-left-style: solid;

border-left-color: orange;

}

#### Сокращённая запись [border]

Пример:

p {

border-width: 1px;

border-style: solid;

border-color: blue;

}

можно объединить в:

p {

border: 1px solid blue;

}

# Ресурсы

1. Справочник CSS. – URL: <http://htmlbook.ru/css>
2. Учебник CSS. – URL: <http://ru.html.net/tutorials/css/>
3. Изучение Веб-разработки – URL: <https://developer.mozilla.org/ru/docs/Learn/CSS>
4. CSS Validation Service. – URL: <http://jigsaw.w3.org/css-validator/validator.html.ru>